Spanning Tree Representation of High Dimensional Data

by

Ding Ding

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Master of Science.

Baltimore, Maryland

May, 2017

Abstract

When analyzing a complex differentiation process with single-cell genomic data, ordering cells in a spanning tree with multiple branches is a useful way to study the dynamic activities of genes along the biological process. Few methods are available for cells from a complex developmental process with branches. We propose a method that uses single-cell RNA sequencing(RNA-Seq) data to construct a spanning tree and orders the cell accordingly. The method first groups cells into clusters. It then randomly samples a cell from each cluster to build a Minimum Spanning Tree(MST). Next,cells are aligned to the tree and ordered based on their projections. Exploring multiple randomly sampled trees with different branching points and cluster numbers is used to identify an appropriate branching structure. Our method can be applied to construct a spanning tree representation for any high-dimensional data set similar to single-cell RNA-Seq data.

Primary Reader: Hongkai Ji

Secondary Reader: Ni Zhao

Acknowledgments

I would like to thank my adviser Hongkai Ji. He is always helpful and approachable. He influenced me not only with his professional attitude towards research, but also his personality. It is my honor to work with him in his group.

I would like to thank my thesis reader Ni Zhao. She gave many useful suggestions for the thesis.She always responds fast with great patience.

I would like to thank my friend Almira Abilova for help with grammar.

I would like to thank the post-doc and graduate students in Hongkai's group, Zhicheng Ji, Weiqiang Zhu, Weixiang Fang, Yifan Zhou, for their advice and help with my thesis.

Last but not least, I always appreciate the support and trust from my parents and younger sister. I love them all the time.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

For mammals, all cells of the body originate from one zygote [1]. Starting from the zygote, each cell needs to choose its fate many times all the way down to choosing a tissue. Some of the cells are programmed to death, such as the negative and positive selection in T cell development [2]. Some of the cells differentiate into a specific terminal stage cell type, such as neuron cell and neuroglia cell [3]. Some cells keep self-renewal capacity. Those cells consistently replenish new cells during metabolism or healing process. Examples include multi-potential hematopoietic stem cell(mHSC) [4] and hepatocyte [5]. Obviously, the developmental process of a cell resembles a typical tree structure with a trunk and multiple branches. To study the structure of the development process, we need to know the relation among cells. For

many other problems, individual objects in a population also have a tree structure.

In reality, we often have measurements for each individual, but do not know the underlying population structure. Computational method is needed to infer the underlying tree structure using measurements of individual objects. For example, consider a mixture of cells with different lineages. In order to study the differentiation process, we can first measure the profile of each cell, and then computationally infer the underlying tree structure that reflects the progressive transition of the measured profiles.

Precise measurement is the cornerstone in any analysis.RNA-Seq [6] is a highly sensitive and accurate tool to study the transcriptome at the nucleotide level. RNA-Seq collected from different conditions can be used to compare the changes in gene expression over time, or differences in gene expression in different groups or treatments. The conventional RNA-Seq measures the average gene expression profiles of all cells in a sample. This average transcriptome works well when all the cells are homogeneous. However, it may fail to catch important transcriptional signals in a heterogeneous cell population [7]. Furthermore, using the average transcriptome to study characteristics of each cell or cell subpopulations can be misleading because of the Simpson's paradox [8].

Advanced single-cell RNA-Seq technique allows one to study cellular heterogeneity at the whole transcriptome level [9,10]. However, computational methods for inferring the biological structure in a cell population from single-cell RNA-Seq data are still

immature. This article develops a spanning tree based method to order single cells according to their underlying biological process. The inferred tree structure can be used to discover individual gene expression changes along the biological process. It may reveal new marker genes or new differentiation path.

1.2 Related Work

Ordering cells by 'pseudo-time', which is a quantitative measure of progress through a biological process, has been studied by multiple authors previously. A supervised learning method, SPD [11], is developed to resolve progression along multiple lineages. However, SPD only works with bulk expression information. For single-cell mass cytometry profile, algorithms like SPADE [11] are developed to arrange cells with lineage relationship. Monocle [12] is the first computational method using single-cell RNA-Seq data to construct the biological order. Monocle treats each cell as a node. It uses dimension reduced single-cell RNA-Seq data to construct a MST. The algorithm reports the transition trajectory of cells along the tree and assigns a pseudo-time to each cell.

Since the MST directly constructed using individual cells is unstable. TSCAN [13] uses a cluster-based MST approach to order cells. This approach improves stability of tree by clustering cells first. It then picks the center point of each cluster to construct the MST and maps cells to the tree to obtain pseudo-time.

Both Monocle [12] and TSCAN [13] focus mainly on one path. For more complex structure of cells, they only provide alternative branches for user to choose with prior knowledge. Another method, Wishbone [14], focuses on more complex developmental pathways with bifurcation. Wishbone is also an unsupervised method that works well on constructing structure with trunk and two branches. It can be applied to both mass cytometry and single-cell transcriptome data. However, Wishbone only allows one bifurcation for the tree structure. SPADE algorithm [15] overcomes this problem by allowing the lineage tree to have multiple branching points. In SPADE, single-cell gene expression data is collected at different time points in the experiment. Thus, the time information of data collection is used to supervise constructing the lineage paths. Without such time information, it is difficult to apply SPADE [15] .

1.3 Algorithm Novelty

In this study, we develop an unsupervised method to infer the underlying tree structure of cells using single-cell RNA-Seq data. We use the inferred tree to estimate pseudo-time and order cells. Unlike previous methods, our method explores a random sample of trees and automatically chooses the best spanning tree structure. It allows the trees to have multiple branches and can describe genes' dynamic activities using smooth branching functions.

Chapter 2

Material and Method

2.1 Overview of the Method

The objective of our method is to construct a spanning tree to represent the structure of a group of objects with high-dimensional measurements. The objects in the group are assumed to have a complex lineage relationship. We have data for each objects to describe its characteristics(e.g. transcriptome). However, the underlying lineage structure is unknown.

Our method provides a way to study the relationship of all objects within the group. Mathematically, each object i is associated with a vector $\boldsymbol{x}_i (i = 1, 2, ..., n)$. The vector is the object's profile. Objects close to each other have similar profiles. In our application, objects are cells, and $\boldsymbol{x}_i$ is the gene expression profile of cell i measured using single-cell RNA-Seq. Each vector $\boldsymbol{x}_i$ contains measurements of all

genes($\sim 20,000$) in the genome.

In order to construct the tree representing the relationship of all cells, we first cluster $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$ into N subgroups. Then we treat each cluster as a node and use the N nodes to construct a MST to serve as the backbone of the lineage structure. The tree consists of a trunk and several branches extended in different directions. With the backbone at hand, we order all objects along the trunk and branches. Each object gets a pseudo-time and branch label on the tree. After we get the tree with all objects mapped, we are able to study how different features of objects change along the tree structure. For example, we can study how a gene changes its expression along the trajectory defined by the tree.

Because one can build many possible trees from objects $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$, we hope to be able to pick a tree as close as possible to the true structure of the objects. For this reason, we only use a subset of the objects to build a tree, and then the tree is evaluated using the remaining objects to see how well one can explain the left-out objects using the tree.

To make the method computationally efficient, we make two assumptions :

(1) The change along the tree is continuous. For example, to study development, we assume that the cells differentiate on a continuous basis.

(2) The tree is binary. In other words, each parent branch is assume to have only two child branches. Although we assume the tree is binary, our method handle the situation where a parent branch is divided into three or more different child branches.

Such multiple branching events can be modeled using two or more close branching points.

2.2 Simulated Data

In order to illustrate and evaluate the method, we simulated a two dimensional data set consisting of 340 single cells. Each cell has GeneA and GeneB expression values generated according to the following steps:

(1) The tree has five branches. It starts from branch1(50 cells) and bifurcates into branch2(100 cells) and branch3. Branch3(50 cells) then bifurcates into branch4(70 cells) and branch5(70 cells). Each cell is labeled with a branch index and a rank on the tree trajectory. For example, the second cell on branch2 is ranked 52 in the pseudo-time trajectory defined by the tree and the third cell on branch3 is ranked 53.

(2) With known rank $x_i, (i = 1, 2, ..., 340)$, branch indicator vector $\boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}, \boldsymbol{b_4}, \boldsymbol{b_5}$ and branching point $B1, B2$ we generated GeneA and GeneB according to the functions below:

$$\begin{aligned} GeneA_i =& a_1 x_i + h_1 x_i^2 + c_1 x_i^3 + d_1 b_{2i}(x_i - B1)_+^3 + e_1 b_{3i}(x_i - B1)_+^3 \\ &+ f_1 b_{4i}(x_i - B2)_+^3 + g_1 b_{5i}(x_i - B2)_+^3 \\ GeneB_i =& a_2 x_i + h_2 x_i^2 + c_2 x_i^3 + d_2 b_{2i}(x_i - B1)_+^3 + e_2 b_{3i}(x_i - B1)_+^3 \\ &+ f_2 b_{4i}(x_i - B2)_+^3 + g_2 b_{5i}(x_i - B2)_+^3 \end{aligned}$$

(3) GeneA, GeneB and rank $\boldsymbol{x}$ are then normalized to interval 0 to 1. Noise $\epsilon_i \sim N(0, 0.04)$ is added to GeneA and GeneB(Figure 2.1A).

As in real data, we use only GeneA and GeneB out of 340 cells to construct a spanning tree based on our method. When the process is done, we can use the original known structure to verify the tree picked from our method for evaluation purposes.

Our method is developed for high-dimensional data set. For simplicity of demonstration, we only simulated two genes in out simulation data. Each gene is a feature. for high-dimensional data set, we first reduce dimensions using methods such as principal component analysis(PCA) [16]. The dimension reduced data are then used to construct trees.

2.3 Data Pre-processing

Assume there are N single cells, $\mathbf{E}_i$ is the gene expression vector of cell $i(i = 1, 2, ..., N)$. The input of our method is the read count of each gene in each cell. We remove genes with no read in any cell. Then we remove any cell if less than 10% of the genes have non-zero count in the cell. The transcripts per million(TPM) is calculated and $\log 2$ transformed to create an expression matrix $\mathbf{E}$ for the retained genes and cells. $\mathbf{E}$ is a $N \times M$ matrix with each cell in a row and each gene in a column.

2.4 Clustering Cells

Since many genes are correlated, we used PCA [16] to transform matrix $\mathbf{E}$ into a lower dimensional matrix $\mathbf{F}$, which is a $N \times K$ matrix with each row representing a cell and each column representing a principal component. Here the PCA is done after scaling each column of $\mathbf{E}$ to have zero mean and unit standard deviation.

The number of principal components is determined using the piecewise linear fit method described in TSCAN [13]. After PCA, we get a $N \times K$ matrix $\mathbf{G}$, where $K \leq M$. Next, we cluster cells into C clusters based on the new matrix $\mathbf{G}$ using model-based clustering function(R package mclust) [17]. We construct the tree using cluster rather than using individual cells. As TSCAN [13], this is because constructing tree directly from cells could make the tree unstable. The tree could be affected easily by noise of individual cells. .

Mclust [17] uses a normal mixture modeling for model-based clustering. It assumes a Gaussian mixture model

$$\Pi_{i=1}^{N} \Sigma_{k=1}^{C} \tau_k \phi_k(\mathbf{G}_i | \mu_k, \Sigma_k)$$

where $\mathbf{G}$ is the data, C is the number of clusters, τ_k is the probability that cell $i (i = 1, 2, ..., N)$ belongs to the kth cluster. The clusters in the model are ellipsoidal, centered at the mean μ_k with covariance matrix Σ_k. By default, Bayesian Information Criterion(BIC) [18] is calculated in Mclust. The model with the optimal BIC is selected among all covariance structures and number of clusters.

The default cluster number in mclust [17], which is up to nine, may be not appropriate for complex structures like a multiple branches spanning tree, as shown on Figure 2.1 and Figure 2.2. When we set the cluster number to 20, the tree built on 20 nodes of the clusters is consistent with the data's structure. The best cluster number chosen by mclust is 6 for the same data set. Although it is enough for the clustering, tree constructed based on six nodes is much simpler than the true structure.

In our algorithm, we let user decide the number of clusters. In addition, we provide a method to evaluate and find the best number of cluster, which is allowed to be larger than nine to better represent data structure.

2.5 Construct Tree

2.5.1 Random Sampling

After clustering the cells into C clusters, we construct a spanning tree that connects all cell clusters. In TSCAN [13], this is done by constructing a MST to connect cluster centers. However, the approach may not accurately identify branching points when the tree has multiple branching points (Figure 2.3).

Instead of using the cluster centers, we randomly sample one point from each cluster, and repeat the process multiple times. Each time we use randomly sampled C points to construct an MST and align all cells to the tree to make the final complete tree. The sampling is done multiple times, and multiple trees are constructed. We

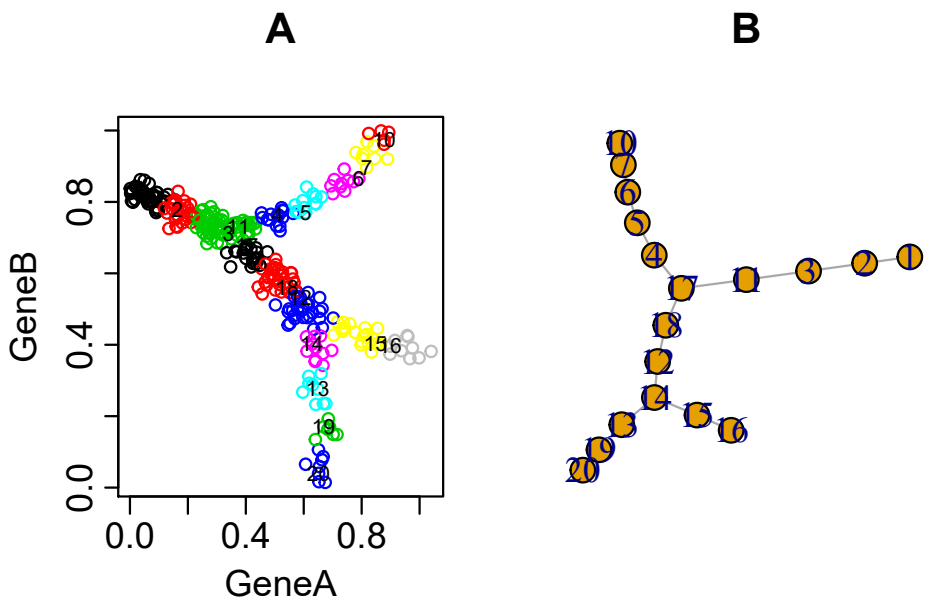

Figure 2.1: Result of given cluster number. Parameter of cluster size equals 20 by user. (A)Each cluster has an unique colour.(B) MST is built with centers of 20 clusters. Each node represents a cluster center. Number in the circle is the cluster id.

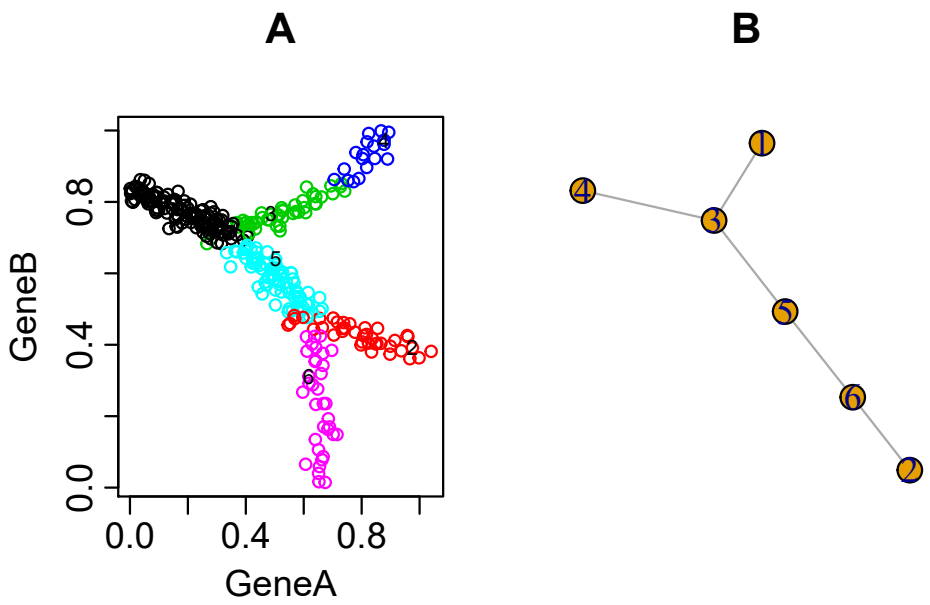

Figure 2.2: Parameter of cluster size chosen by Mclust [17] by default. (A)Each cluster has an unique colour.(B) MST is built with centers of 6 clusters. Each node represents a cluster center. Number in the circle is cluster id.

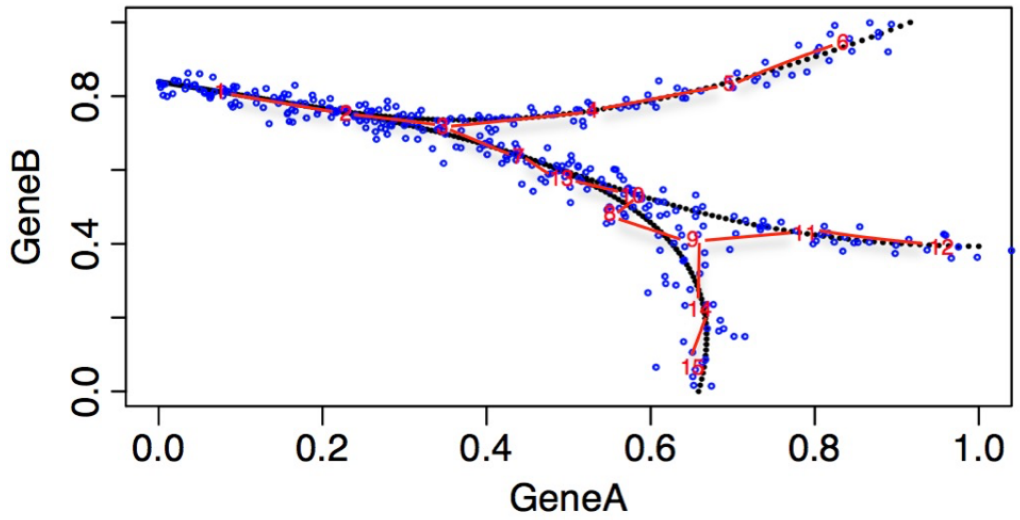

Figure 2.3: MST built with cluster centers. Blue points are true value of GeneA and GeneB. Numbers in red are centers of each cluster. Red lines between numbers are backbone of MST

will then use the method described below to select the best tree among all sampled trees.

2.5.2 Spanning Tree

Here we use the C points randomly sampled from each cluster and a minimum spanning tree function in R package on graph [19] to build an MST with undirected and weighted setting.

A spanning tree is a subset of edges of a connected edge-weighted undirected graph that connects all the vertices together, without any cycles. An MST is the spanning tree with minimum possible total edge weight. The vertices are the C nodes and the weight is the Euclidean distance between each two points on C.

The backbone of tree with C nodes provides a rough trajectory. However, MST itself does not have a direction. There are two possible ways to determine the start of the pseudo-temporal trajectory:

(1) Users have some prior knowledge of some marker genes, which can be used to identify the start cluster;

(2) There are some cells which are the blast cells in the experiment, like the data we used in the result part. Known blast cells are used from the beginning to conduct a differential process. Cluster 1 in Figure 2.3 is picked out this way, since the percentage of original cells in this cluster is higher than in the other clusters.

Beginning from the start cluster , the algorithm will go through the whole tree including all branches to determine the sequence of clusters and cells on each path. For example, in Figure 2.3, the main trunk is $1 \rightarrow 2 \rightarrow 3$, which is treated as branch1. Branch2 is $3 \rightarrow 4 \rightarrow 5 \rightarrow 6$. Branch3 is $3 \rightarrow 7 \rightarrow 13 \rightarrow 10 \rightarrow 8 \rightarrow 9$. Branch4 is $9 \rightarrow 11 \rightarrow 12$ and branch5 is $9 \rightarrow 14 \rightarrow 15$. Branch 1 has child branches 2 and 3. Branch 3 has parent branch 1 and child branches 4 and 5. Each node in the tree bone is represented as $\mathbf{C}_i (i = 1, 2, ..., C)$. A path is defined as the whole sequence of clusters from start cluster to end cluster.

2.6 Order Cells Along The Tree

With the structure of the tree from top to bottom on cluster level and each branch listed separately, all cells are aligned to the tree we constructed. For each cell $\mathbf{G}_i (i = 1, 2, 3, ..., N)$, firstly, we found nearest two node $\mathbf{C}_j$ and $\mathbf{C}_h$. If $\mathbf{C}_j$ and $\mathbf{C}_h$ are on the same branch(for example $j = 14$ and $h = 9$ in Figure 2.3), we will align the cell $\mathbf{G}_i$ to the edge $\mathbf{v}_{jh} = \mathbf{C}_j - \mathbf{C}_h$ if h precedes to j in the branch or $\mathbf{v}_{jh} = \mathbf{C}_h - \mathbf{C}_j$(in the example $j = 14$ and $h = 9$, $\mathbf{v}_{14,9} = \mathbf{C}_{14} - \mathbf{C}_9$). The projection of cell $\mathbf{G}_i$ to $\mathbf{v}_{jh}$ is defined by the inner product as in TSCAN [13]

$$\frac{\mathbf{v}_{jh}^T \mathbf{G}_i}{\| \mathbf{v}_{jh} \|}$$

where $\| . \|$ is the l^2-norm of a vector.

If $\mathbf{C}_j$ and $\mathbf{C}_h$ are not in the same branch(for example $j = 14$ and $h = 8$), we will keep the nearest node $\mathbf{C}_j$, and find the third or fourth, or even further node along the nearest node list until we get $\mathbf{C}_l$ that $\mathbf{C}_j$ and $\mathbf{C}_l$ in the same branch. The edge to project is $\mathbf{v}_{jh} = \mathbf{C}_j - \mathbf{C}_l$(when l precedes j in the branch) and the projection function above is used to get the projection value of cell i. Here l is defined as a preceding node of j if l appears earlier than j in pseudo-time along the branch.

After all the cells are projected to the tree, we use the following criteria to rank the cells :

(1) We rank the cells on branch level. Within each branch, we rank the cells based on the preceding node of the edge the cells are projected to(in example above, when

$j = 14$ and $h = 9$, and $\mathbf{v}_{14,9} = \mathbf{C}_{14} - \mathbf{C}_9$,$\mathbf{C}_9$ is the preceding node in the edge and for the cell). All cells with preceding node 9 have smaller ranks than cells with preceding node 14.

(2) For cells with the same preceding node on the same branch, the projected values are used to rank the cells.

(3) After ranking cells on each branch, we re-rank the cells in spanning tree level. In general, parent branch always has smaller rank than child branches. For example, all cells in branch 1 have smaller rank than the cells in branch 2. The trunk, or branch1 has the same rank as branch level, while for other branches, each child branch is ranked after its parent branch. For example, in Figure 2.3, the branch1 has 100 cells ranked from 1 to 100. The 85 cells in branch2 are ranked from 101 to 150, and the 30 cells in branch3 are ranked from 101 to 130. The 50 cells in branch4 are ranked from 131 to 170, and the 35 cells in branch5 are ranked from 131 to 165.

Now, each cell i has two labels (r_i and b_i), r_i is the rank of the cell in the tree and b_i is the branch the cell belongs to. In the example above, there are 100 cells in branch1, the second cell j in branch2 would be cell(r_j=102,b_j=2), while the second cell h in branch3 would be cell(r_h=102, b_j=3). For visualization and calculation convenience, we rescale the r_i to range [0,1] with the function $(r_i - min(\mathbf{r}))/(max(\mathbf{r}) - min(\mathbf{r}))$.

2.7 Parametric Modelling of Gene Expression Dynamics

After constructing the tree using matrix $\mathbf{G}$ with N cells and the first M principal components, we obtain the rank $\mathbf{r}$ and branch of all cells, which will then be treated as the latent pseudo-time of the cells. If we plot each dimension(principal component) $\mathbf{G}_{.j}$ in y-axis and $\mathbf{r}$ in the x-axis, we could see a pattern similar to spanning tree in each dimension with the same branching number and branching pseudo-time.For example, in Figure 2.4, simulated GeneA and GeneB's expressions have known rank(scaled within [0,1]) as shown. Estimated rank of cells are acquired now, and we use the rank as latent pseudo-time. Here, we plot genes' expression in true time and in pseudo-time respectively. We can see from the plot that the expression curves estimated using pseudo-time can represent the expression changes in the real time.

Next, we fit a linear regression of each dimension $\mathbf{G}_{.j}$ with basis combination from B-spline [20] of $\mathbf{r}$.Each dimension is a gene in simulated data for simplicity. For real RNA-Seq data, it is a principal component.

We use simulated data as an example to illustrate the linear model on each principal component. First, we get the basis of $\mathbf{r}$ with bs function in R package splines. The knots for B-spline are t equally separated points from 0 to 1 where the default we use in the algorithm is 15. Users can change t and even choose to set knot manually or differently among each intervals marked by branching points. The basis matrix $\mathbf{B}$

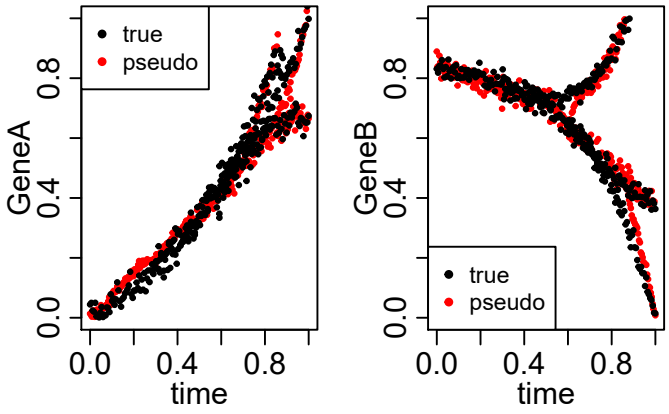

Figure 2.4: Comparing each gene's expression level along estimated pseudo-time to true time. Red points are gene expression on estimated pseudo-time for all cells. Black points are gene expression on true time used for generation of simulated data.

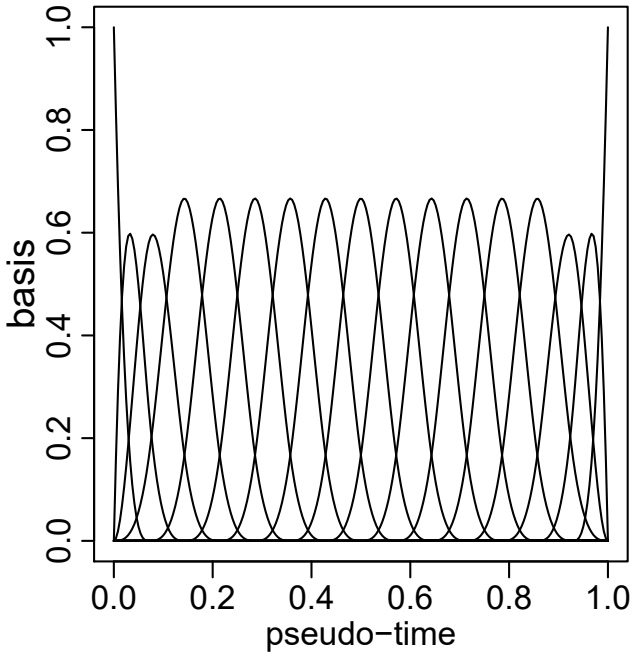

Figure 2.5: 17 Basis from B-spline of pseudo-time. Knots for the B-spline are set evenly between 0 and 1. The number of basis depends on user's choice. 17 is the default number.

has N rows and $t + 2$ columns. Each row is a cell. Each column is a basis. There are 17 basis in the example,as shown in Figure 2.5.

Here we use GeneA in the example as the response in the model. As Figure 2.6 shows, within the pseudo-time range([0,1]), branches can be separated by branching points(branching point 1=0.602 for branch1,2,3, and branching point 2= 0.855 for branch3,4,5). Each branch is modeled only with the basis beginning within the

branch.

In the example, Branch1 is modeled with basis 1 to 11(defined as $\mathbf{b}^{(1)}$). Branch3 is modeled with basis 12 to 15(defined as $\mathbf{b}^{(2)}$) and Branch4 is modeled with basis 16 to 17(defined as $\mathbf{b}^{(3)}$). We define expression Gene A to cell j is $y_{b,j}$,$b = 1, 2, 3, 4, 5$ for branch, $j = 1, 2, ..., N$ for cell number. The matrix form of linear regression for Gene A is

$$
\begin{bmatrix} y_{1,1} \\ \dots \\ y_{1,n_1} \\ y_{2,n_1+1} \\ \dots \\ y_{2,n_2} \\ y_{3,n_2+1} \\ \dots \\ y_{3,n_3} \\ y_{4,n_3+1} \\ \dots \\ y_{4,n_4} \\ y_{5,n_4+1} \\ \dots \\ y_{5,N} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{b}_1^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{b}_{n_1}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{b}_{n_1+1}^{(1)} & \mathbf{b}_{n_1+1}^{(2)} & \mathbf{b}_{n_1+1}^{(3)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{b}_{n_2}^{(1)} & \mathbf{b}_{n_2}^{(2)} & \mathbf{b}_{n_2}^{(3)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{b}_{n_2+1}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{b}_{n_2+1}^{(2)} & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{b}_{n_3}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{b}_{n_3}^{(2)} & \mathbf{0} & \mathbf{0} \\ 1 & \mathbf{b}_{n_3+1}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{b}_{n_3+1}^{(2)} & \mathbf{b}_{n_3+1}^{(3)} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots & \mathbf{0} \\ 1 & \mathbf{b}_{n_4}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{b}_{n_4}^{(2)} & \mathbf{b}_{n_4}^{(3)} & \mathbf{0} \\ 1 & \mathbf{b}_{n_4+1}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{b}_{n_4+1}^{(2)} & \mathbf{0} & \mathbf{b}_{n_4+1}^{(3)} \\ \dots & \dots & \dots & \dots & \dots & \mathbf{0} & \dots \\ 1 & \mathbf{b}_{n_5}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{b}_{n_5}^{(2)} & \mathbf{0} & \mathbf{b}_{n_5}^{(3)} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_{26} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \dots \\ \epsilon_{n_1} \\ \epsilon_{n_1+1} \\ \dots \\ \epsilon_{n_2} \\ \epsilon_{n_2+1} \\ \dots \\ \epsilon_{n_3} \\ \epsilon_{n_3+1} \\ \dots \\ \epsilon_{n_4} \\ \epsilon_{n_4+1} \\ \dots \\ \epsilon_N \end{bmatrix}
$$

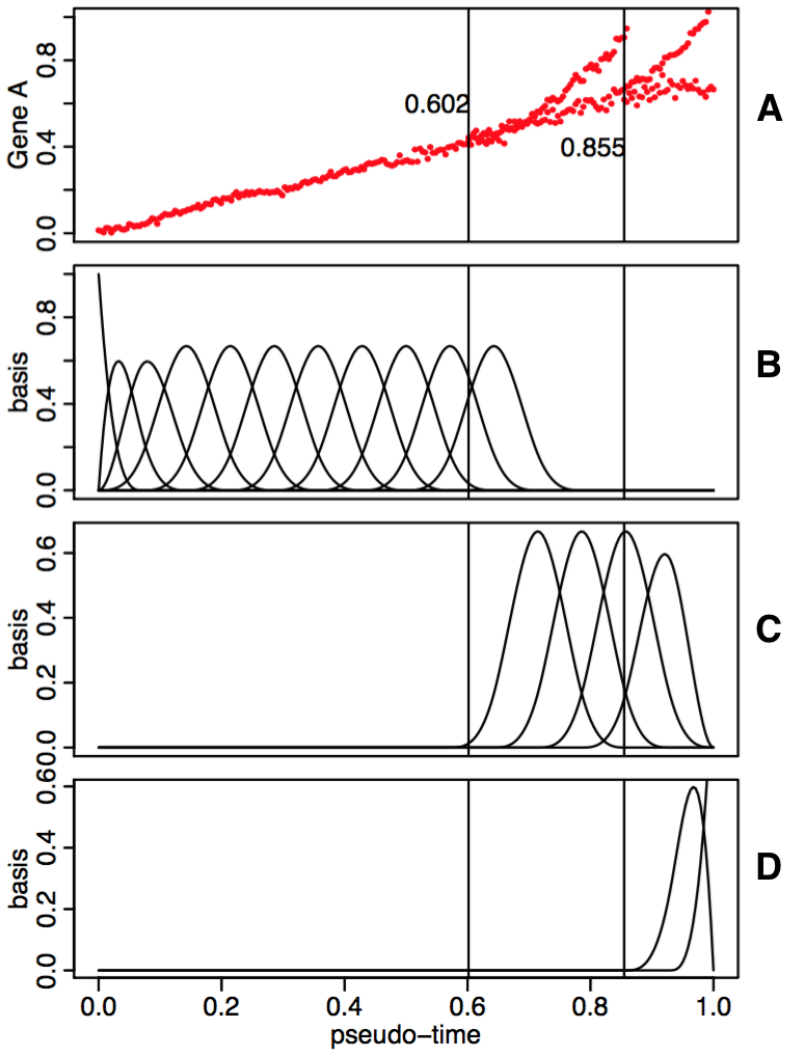

Figure 2.6: Linear model for GeneA on B-Spline basis. (A)GeneA expression on pseudo-time. Vertical lines are branching points at 0.602 and 0.855. (B)The basis used to model the branch1(before 0.602). (C)The basis used to model the branch3(between 0.602 and 0.855). (D)The basis used to model the branch4 and branch5(after 0.855).

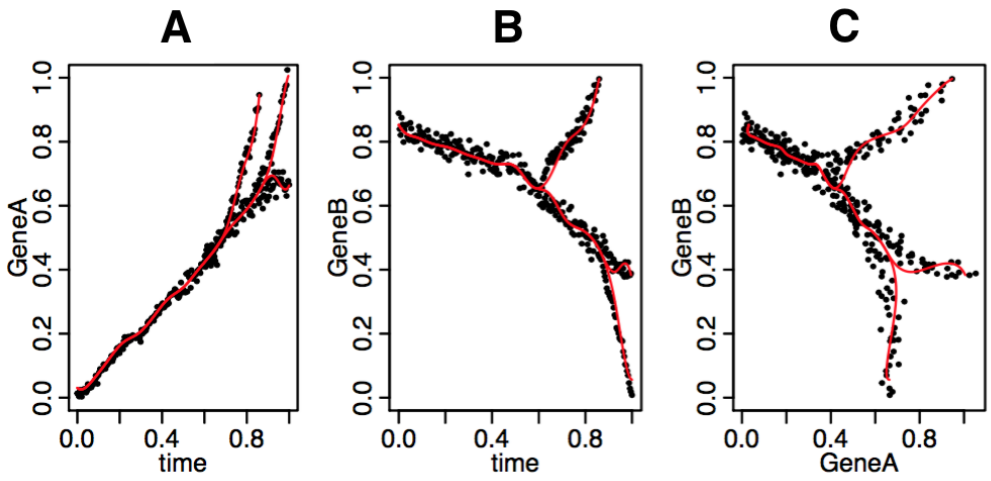

Figure 2.7: Fitted Lines(red) of GeneA(A) and GeneB(B) on estimated pseudo-time(black points).(C) Red line is fitted value on estimated pseudo-time. Black points are simulated data.

where $\epsilon_i \sim N(0,\sigma^2)$, (i=1,2,...,N).

We fit the linear regression to each gene in the example or each principal component in any data set. The result of the fitted line for GeneA and GeneB are shown in Figure 2.7. Furthermore, to improve the prediction precision , we used Ridge regression [21](cv.glm function which picks the λ for the penalty with cross validation automatically in R package glmnet is used). The ridge regression used has the same design matrix shown above but with penalty added to the parameters $\boldsymbol{\beta}$ during estimation.

2.8 Tree Evaluation

After building multiple trees using different cluster numbers and randomly sampled points as a backbone from each cluster, we need a criterion to evaluate each tree. This criterion is used to identify the tree that best represents the data.

2.8.1 Cross Validation

At the beginning of the whole algorithm, after processing the data, we randomly select 70% of all cells (training data set $\mathbf{C}_{N_1 \times M}, N_1 = 0.7 \times N$) for tree building and model fitting, and keep the remaining 30% (testing data set $\mathbf{V}_{N_2 \times M}, N_2 = 0.3 \times N$) to check the model. The training and testing data sets are the same for all the constructed trees.

2.8.2 Marginal Probability Score

We fit a model for each column of $\mathbf{G}_{N \times M}$(principal components) on pseudo-time. For a given cell i in $\mathbf{V}_{N_2 \times M}$, y_{im} is the cell i's value on m_{th} dimension. If the prior probability that y_{im} comes from pseudo-time t is $P_T(t)$, then we can get

$$P_Y(y_{im}) = \int_t P_{Y|T}(y_{im}|t) P_T(t) dt$$

Considering the fact that we do not know $P_T(t)$, we assume that $P_T(t)$ has a uniform distribution and that y_{im} has the same probability to come from any pseudo-time

t. Thus, we have N_1 points (t_j, Y_j) from the model fitted on dimension m, t_j in the pseudo-time and Y_j a fitted value that equals to $E[y|t_j], j = 1, 2, ..., N_1$. Based on the linear regression model fitted above, we have $y|t_j \sim N(Y_j, \sigma^2)$, and σ^2 estimated by mean square error(MSE) of the model on dimension m. Further, we use these N_1 points (t_j, Y_j) to calculate the marginal probability of $P_Y(y_{im})$

$$P_Y(y_{im}) = \frac{1}{N_1} \sum_{j=1}^{N_1} P_{Y|T}(y_{im}|t_j)$$

For all N_2 cells in testing data, we define marginal likelihood score(MLS) in dimension m

$$MLS_m = \log_2 \prod_{i=1}^{N_2} P_Y(y_{im}) = \sum_{i=1}^{N_2} \log_2 P_Y(y_{im})$$

And the total MLS is

$$MLS = \sum_{m=1}^{M} MLS_m = \sum_{m=1}^{M} \sum_{i=1}^{N_2} \log_2 P_Y(y_{im})$$

The tree with larger MLS is thought to better represent the data structure.

Chapter 3

Result

3.1 Analysis of Early Mouse Embryonic RNA-Seq Data Set

3.1.1 Data Set Description

We used the data set from Jang et al. [22] for analysis in this paper. The specific mouse embryonic stem(mES) cells were exposed to one or combinations of four molecules in a sequence to mimic a neuron early development process. First, mES cells at day0 were exposed to PD0325901 for 2(day1 and 2) days, or CHIR99021 and Activin A for 3 days(day1 to day3). Cells were collected every 24 hours and media was replenished every 48 hours. Second, cells exposed to PD0325901 were further ex-

posed to either hBmp4 or LDN193189 for another two days(day3 and day4). Third, cells exposed to CHIR99021 and Activin A were also further exposed to either hBmp4 or LDN193189 for another two days(day4 and day5). Cells were also collected every 24hr during second and third stages. It was believed that cells exposed to PD0325901 would give rise to ectodermal lineages. Cells exposed to CHIR99021 and Activin A would give rise to mesendodermal, which further differentiated into endoderm with LDN and mesoderm with hBmp4. All the cells were labeled by the day of collection and the sequence of used molecules, for example, day3/PD0325901/hBmp4. The labels could be used to check the result of our method.

More details of cell cultured was provided in the supplement material of Jang et al. [22]. The design of experiment was to obtain several differentiated cell populations from the same stem cell. Cells at the end of differentiation were expected to be either neural/non-neural ectoderm, definitive endoderm, or mesoderm. Actually, the experiment intended to mimic and construct a lineage relationship from mES cell to differentiated cell subtypes.

It was reasonable to believe that most of the cells collected at different time points followed a sequence from being undifferentiated to becoming differentiated. Cells collected from the same plate should be enriched as a certain subtype. However, as mentioned and confirmed in Jang's experiments [22] by mass cytometry, some cells collected during day4 or day5 remained as stem cells, while some cells from day1 were well differentiated.

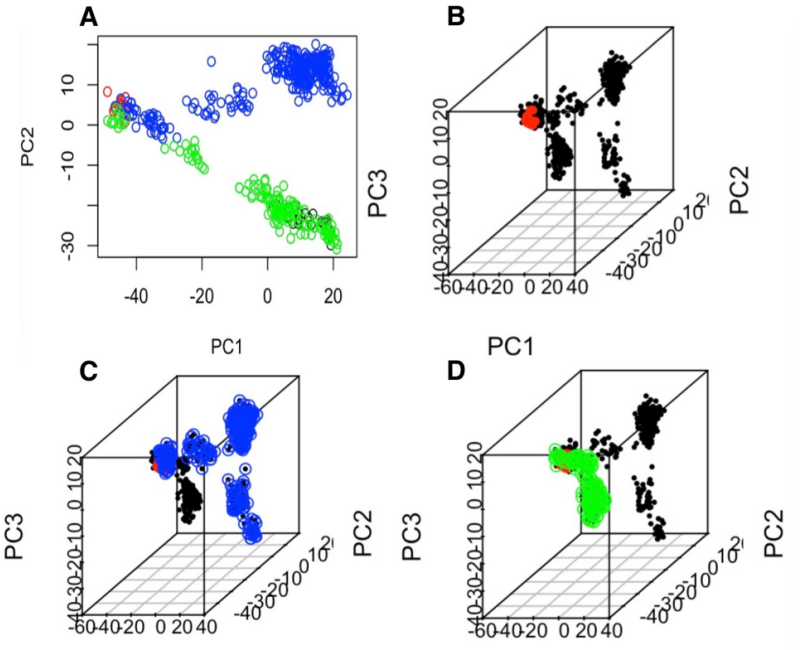

Figure 3.1: Separation of cells on PCs. (A) Cells were separated into two branches on PC1 and PC2. (B,C,D) Cells were further separated into more branches on PC1 to PC3, with different cells representing stem cell and subtypes stimulated by different molecules

3.1.2 Analysis Process

In this experiment we had an RNA-Seq data from 493 single cells. The reads of each cell were around 20,000, which were randomly sampled by Jang [22] from all reads. We applied our method to all genes except the ones without counts in any of the cells(in Jang's article, they only used 283 out of the 493 cells). Exploratory analysis of the cells on different PCs are shown on Figure 3.1. The cells in red were from the start cluster, which consisted mostly of mES cells. Cells in blue consisted mostly of PD0325901 treated cells, while cells in green mostly came from CHIR99021 and Activin A treated cells. Two branches were generated from the mES trunk.

3.1.3 Analysis Result

The first four PCs were used to construct the tree. We used different cluster numbers for C from 10 to 20. For each C, we randomly sampled and constructed 200 trees. The structure of selected best tree(highest MLS) is shown on Figure 3.2. Only the cell labels of stem cell line were used for defining the starting cluster(defined as the cluster with highest percentage of stem cell), other cells' labels included stimulation molecules were concealed. For most marker genes for each development stage, their expression trends along the cells' trajectory on the tree were accurately recovered from our algorithm.

Figure 3.2 shows the cluster level backbone of the constructed tree; all the cells were aligned to the tree's backbone. In order to check the accuracy of the best tree in representing the cell development process, we used known marker genes or transcription factors in each stage of the mES differentiation process. We obtained these marker genes' expression value based on the cells' rank within the tree with all branches. The expression trends shown were marked by each differentiation pathway.

In the experiment, the differentiation process started from certain mES cell line. Figure 3.3 reveled expression of four markers, Klf4, Jarid2, Esrrb and Klf5 along the pseudo-time of branches in the tree. The four genes were markers of undifferentiated naive pluripotent cells [23, 24]. The pseudo-time here was the rank of the cells(scaled to interval [0,1]). The expression trend of Klf4 along four different paths of the tree were shown by different colors. The Klf4 had high expression level at the beginning

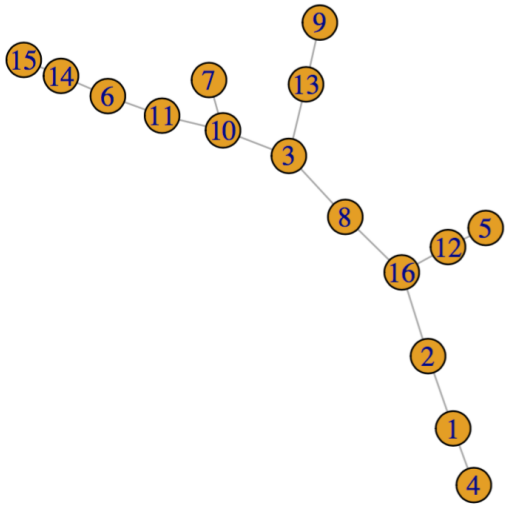

Figure 3.2: Cluster level structure of best tree. It is the backbone of the tree with each points in one cluster. Each circle is a cluster. Numbers in each circle represent the cluster id.

Table 3.1: Main paths of the tree

Path	Cluster Sequence from Start to End
1	$9 \to 13 \to 3 \to 8 \to 16 \to 2 \to 1 \to 4$
2	$9 \to 13 \to 3 \to 8 \to 16 \to 12 \to 5$
3	$9 \to 13 \to 3 \to 10 \to 11 \to 6 \to 14 \to 15$
4	$9 \to 13 \to 3 \to 10 \to 7$

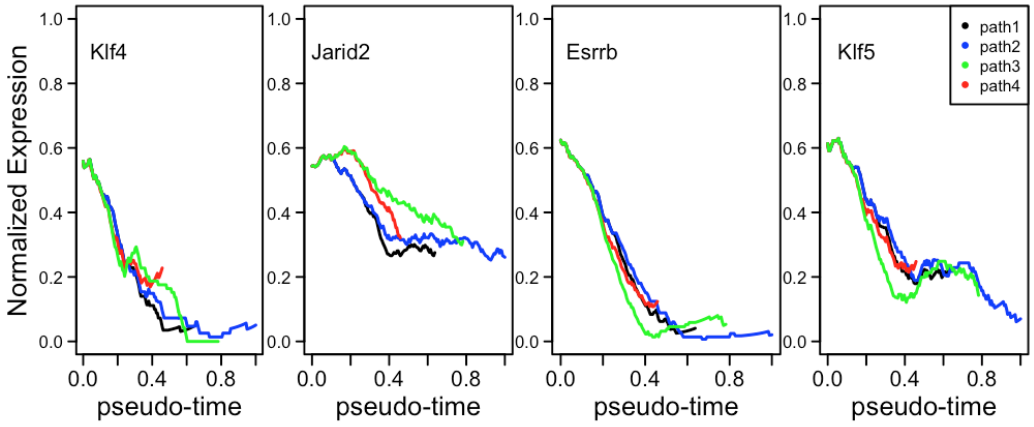

Figure 3.3: Marker genes expression of naive pluripotent(ES) along pseudo-time. For each gene, expressions of four differentiation pathways are marked separately.

of the tree and decreased all the way along all four paths. This result is consistent with the expression pattern of Klf4 along the differentiation process of mES cell. From the expression pattern of Jarid2, there was a common trunk of four paths in the beginning, which then bifurcated into two branches. Path3 and path4 shared a common branch before they bifurcated. The same was true for path1 and path2 that shared a common path before finally differentiating into two lineages. The pattern of the Jarid2 expression was consistent with the structure of the spanning tree. For the other three Klf4, Esrrb and Klf5, the expression levels were not distinguishable along four paths, which was reasonable since marker of naive cells are all mildly expressed along the differentiation processes.

The naive pluripotent cells went through a primed pluripotent epiblast stage before differentiating into either bipotent entoderm or mesendoderm stage [25]. Figure 3.4 gave the expression of Bptf,Cbx1,Otx2 and Sox2 along the pseudo-time. Bptf,

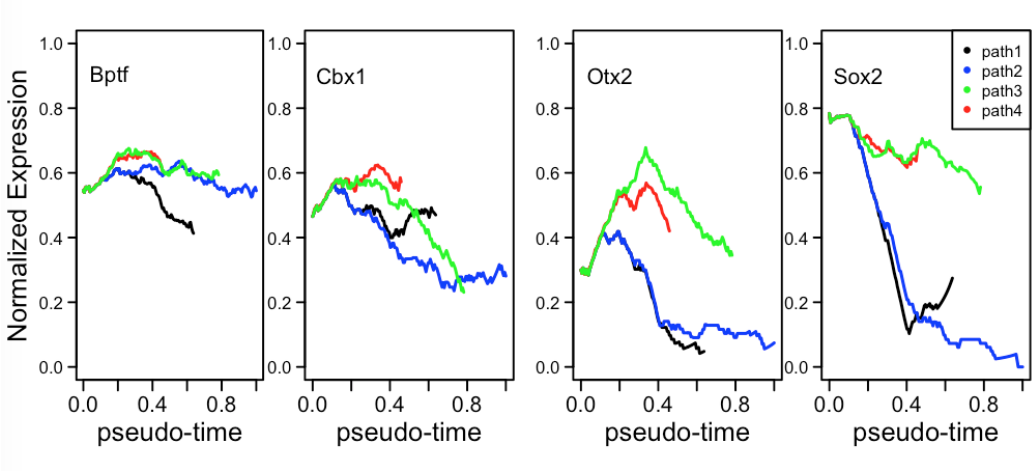

Figure 3.4: Marker genes expression of primed pluripotent epiblast along pseudo-time. For each gene, expressions of four differentiation pathways are marked separately.

Cbx1 and Otx2 are marker genes of primed pluripotent epiblast stages [25–27]. The expression trend from our algorithm showed low expression at the beginning, increasing in the trunk and early part of branches and decreasing along further branches.It was consistent with the expression of marker genes of primed pluripotent epiblast stage. We predicted that cells aligned to the early trunk before the first branching point would belong to the primed pluripotent epiblast stage. Sox2 is thought to be a marker gene for naive and primed pluripotent epiblast [28]. The expression pattern that can be observed from our algorithm demonstrated the high expression in the early stage, and a differential expression level between common part of path1 and path2 and common part of path3 and path4, which were the two branches discussed below.

After the primed pluripotent epiblast stage, the cells were stimulated to differentiate into two subtypes. We used the marker genes Eras, Sez6, Stmn3,Stmn4 of bipotent

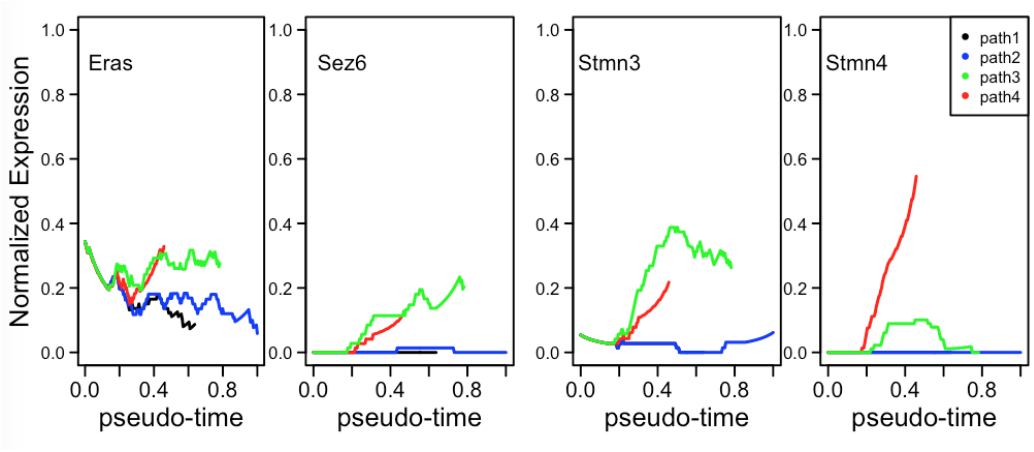

Figure 3.5: Marker genes expression of bipotent ectoderm along pseudo-time. For each gene, expressions of four differentiation pathways are marked separately.

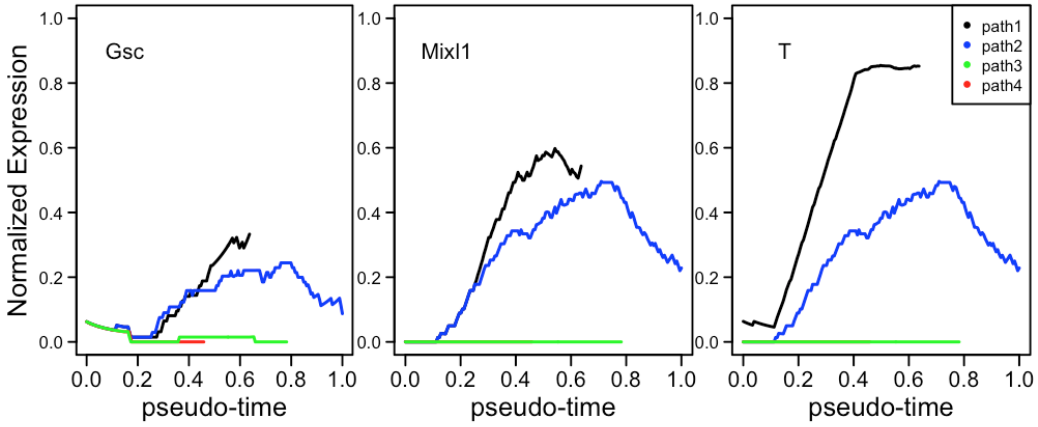

Figure 3.6: Marker genes expression of mesendoderm along pseudo-time. For each gene, expressions of four differentiation pathways are marked separately.

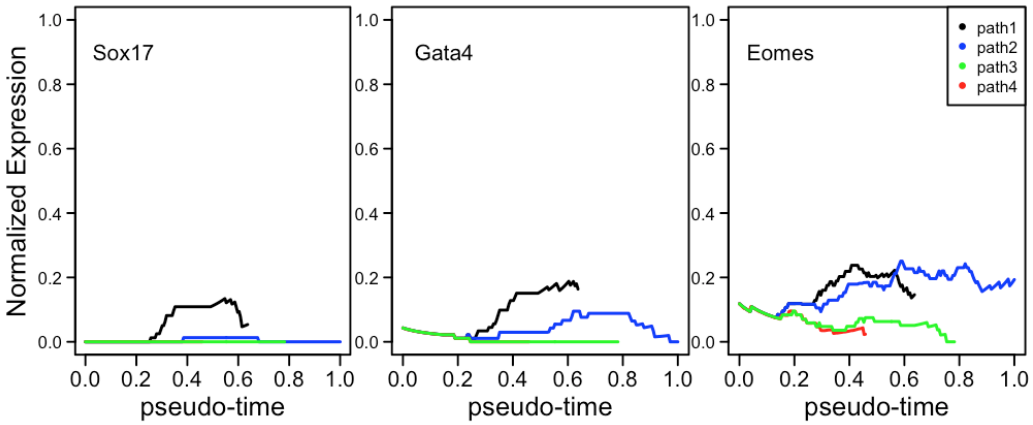

Figure 3.7: Marker genes expression of mesoderm along pseudo-time. For each gene, expressions of four differentiation pathways are marked separately.

ectoderm stage and marker genes GSC, Mixl1,T of mesendoderm stage [29–32]. Comparison of the marker gene expression pattern between these two stage was shown in Figure 3.5 and Figure 3.6. For markers of bipotent ectoderm stage, expression level by cells rank from our method demonstrated higher expression along path3 and path4. For markers of mesendoderm cells stage, expression level by cells rank from our algorithm showed higher expression along path1 and path2. The obvious difference of expression pattern between markers of two groups helped us to predict the path1 and path2 as mesendoderm branch, path3 and path4 as bipotent ectoderm, both of which came from primed pluripotent epiblast. The result from our algorithm was verified by the labels of biological process designed in the experiment.

Mesendoderm cells would further develop into mesoderm and endoderm cells, which were totally different subtypes [33]. We still used the expression of marker genes of mesoderm(Sox17,Gata4 and Eomes) [34–36] based on the tree we built, in which we

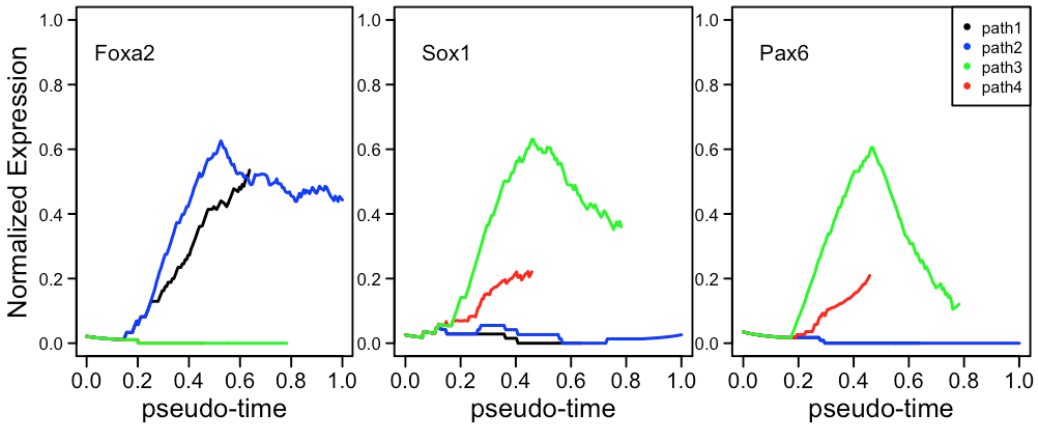

Figure 3.8: Marker genes expression of endoderm and ectoderm along pseudo-time. For each gene, expressions of four differentiation pathways are marked separately.

tried to match the biological process for verification of the accuracy of algorithm. As seen in Figure 3.7, path1 and path2 were associated with high expression especially at the end of each path compared to path3 and path4. Furthermore, Sox17 only had expression in path1 but not path2, which are both bifurcated from the mesendoderm trunk. Also, Gata4 and Eomes had much higher expression level in path1 compared to path2. We predicted that path1 developed into mesoderm and path2 developed into endoderm. This again was validated by the expression level of endoderm stage marker gene Foxa2, which is shown in Figure 3.8. Path2 had a much higher expression level than path 1 which was predicted as endoderm from our method.

This experiment also produced an additional finding. For any two branches generated from the same trunk, the length of the branch was linked to the number of cells aligned to the branch, because our pseudo-time here was the rank of the cells along the tree. Sox1 and Pax6 were marker genes of ectoderm(Figure 3.8), which verified

path3 as ectoderm.While for the neuro crest cell, which was a branch from ectoderm, we did not see obvious pattern of marker gene of neuro crest cells in the path4. Since the bipotent ectoderm differentiated into mainly ectoderm with other subtypes like neuron crest and epidermis, we predicted the path 4 to be a combination of neuron crest and epidermis, which could not be further distinguished due to the limitation of total cell number in the path within this data set. We could see from the figure with Sox1 and Pax6, path4 after the second branch was really short which stand for less cells in this path.

Chapter 4

Conclusion

In this article, we developed an unsupervised algorithm to construct a spanning tree to represent single-cell RNA-Seq data along a biological process. Our algorithm demonstrated an ability to deal with single cells that have complex lineage structure. For a transcriptome data set of single cells with unknown structure, our algorithm can pick up the best structure ranging from a single linear path to a more complex spanning tree. The algorithm also provides pseudo-time and branch information of each cell.

The performance of our method is evaluated in two ways. First, in the simulated data set, the constructed spanning tree and the gene expression along pseudo-time for each dimension are shown to be consistent with the original patterns used to generate the process. Second, for the real mES data set in the results, known marker genes from different paths in the differentiation process are used for evaluation. The

expression trend of these genes along the tree that we constructed from the algorithm successfully matched the true expression trend as expected. Both of these methods show the reliability of the tree obtained from our algorithm for unknown gene expression analysis.

Although our method is developed and evaluated using on RNA-seq data in this article, it may be applied to other problems with the data having a spanning tree structure.

In the future, the method can be improved in several ways. First, the method currently only allows up to three bifurcations of the spanning tree. The future work and research would implement the algorithm to allow any number of bifurcations. In principle, this generalization is straightforward, but we need to modify the codes to realise it. Second, the pseudo-time we get from MST may not be the optimal pseudo-time, and we could update the pseudo-time for each cell iteratively until convergence to get the best pseudo-time for subsequent analyses. Third, a graphic user interfere may be developed for user's convenience.

Bibliography

[1] M. L. Condic, "Totipotency: what it is and what it is not," *Stem cells and development*, vol. 23, no. 8, pp. 796–812, 2013.

[2] C. A. Janeway, "Thymic selection: two pathways to life and two to death," *Immunity*, vol. 1, no. 1, pp. 3–6, 1994.

[3] K. S. Saladin and L. Miller, *Anatomy & physiology*. WCB/McGraw-Hill, 1998.

[4] A. Birbrair and P. S. Frenette, "Niche heterogeneity in the bone marrow," *Annals of the New York Academy of Sciences*, vol. 1370, no. 1, pp. 82–96, 2016.

[5] S. Kawasaki, M. Makuuchi, S. Ishizone, H. Matsunami, M. Terada, and H. Kawarazaki, "Liver regeneration in recipients and donors after transplantation," *The Lancet*, vol. 339, no. 8793, pp. 580–581, 1992.

[6] R. D. Morin, M. Bainbridge, A. Fejes, M. Hirst, M. Krzywinski, T. J. Pugh, H. McDonald, R. Varhol, S. J. Jones, and M. A. Marra, "Profiling the hela

s3 transcriptome using randomly primed cdna and massively parallel short-read sequencing," *Biotechniques*, vol. 45, no. 1, p. 81, 2008.

[7] A. Raj, C. S. Peskin, D. Tranchina, D. Y. Vargas, and S. Tyagi, "Stochastic mrna synthesis in mammalian cells," *PLoS Biol*, vol. 4, no. 10, p. e309, 2006.

[8] E. H. Simpson, "The interpretation of interaction in contingency tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 238–241, 1951.

[9] P. Dalerba, T. Kalisky, D. Sahoo, P. S. Rajendran, M. E. Rothenberg, A. A. Leyrat, S. Sim, J. Okamoto, D. M. Johnston, D. Qian *et al.*, "Single-cell dissection of transcriptional heterogeneity in human colon tumors," *Nature biotechnology*, vol. 29, no. 12, pp. 1120–1127, 2011.

[10] A. R. Wu, N. F. Neff, T. Kalisky, P. Dalerba, B. Treutlein, M. E. Rothenberg, F. M. Mburu, G. L. Mantalas, S. Sim, M. F. Clarke *et al.*, "Quantitative assessment of single-cell rna-sequencing methods," *Nature methods*, vol. 11, no. 1, pp. 41–46, 2014.

[11] P. Qiu, A. J. Gentles, and S. K. Plevritis, "Discovering biological progression underlying microarray samples," *PLoS Comput Biol*, vol. 7, no. 4, p. e1001123, 2011.

[12] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn, "The dynamics and reg-

ulators of cell fate decisions are revealed by pseudotemporal ordering of single cells," *Nature biotechnology*, vol. 32, no. 4, pp. 381–386, 2014.

[13] Z. Ji and H. Ji, "Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis," *Nucleic acids research*, vol. 44, no. 13, pp. e117–e117, 2016.

[14] M. Setty, M. D. Tadmor, S. Reich-Zeliger, O. Angel, T. M. Salame, P. Kathail, K. Choi, S. Bendall, N. Friedman, and D. Pe'er, "Wishbone identifies bifurcating developmental trajectories from single-cell data," *Nature biotechnology*, vol. 34, no. 6, pp. 637–645, 2016.

[15] M. J. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Machine learning*, vol. 42, no. 1, pp. 31–60, 2001.

[16] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[17] C. Fraley and A. E. Raftery, "Mclust version 3: an r package for normal mixture modeling and model-based clustering," DTIC Document, Tech. Rep., 2006.

[18] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[19] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal, Complex Systems*, vol. 1695, no. 5, pp. 1–9, 2006.

[20] G. D. Knott, *Interpolating cubic splines*. Springer Science & Business Media, 2012, vol. 18.

[21] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[22] S. Jang, S. Choubey, L. Furchtgott, L.-N. Zou, A. Doyle, V. Menon, E. B. Loew, A.-R. Krostag, R. A. Martinez, L. Madisen *et al.*, "Dynamics of embryonic stem cell differentiation inferred from single-cell transcriptomics show a series of transitions through discrete cell states," *eLife*, vol. 6, p. e20487, 2017.

[23] J. Kim, J. Chu, X. Shen, J. Wang, and S. H. Orkin, "An extended transcriptional network for pluripotency of embryonic stem cells," *Cell*, vol. 132, no. 6, pp. 1049–1061, 2008.

[24] R. A. Young, "Control of the embryonic stem cell state," *Cell*, vol. 144, no. 6, pp. 940–954, 2011.

[25] J. Nichols and A. Smith, "Naive and primed pluripotent states," *Cell stem cell*, vol. 4, no. 6, pp. 487–492, 2009.

[26] T. Goller, F. Vauti, S. Ramasamy, and H.-H. Arnold, "Transcriptional regulator bptf/fac1 is essential for trophoblast differentiation during early mouse development," *Molecular and cellular biology*, vol. 28, no. 22, pp. 6819–6827, 2008.

[27] Q. Zhou, H. Chipperfield, D. A. Melton, and W. H. Wong, "A gene regulatory

network in mouse embryonic stem cells," *Proceedings of the National Academy of Sciences*, vol. 104, no. 42, pp. 16 438–16 443, 2007.

[28] A. Rizzino, "Sox2 and oct-3/4: a versatile pair of master regulators that orchestrate the self-renewal and pluripotency of embryonic stem cells," *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, vol. 1, no. 2, pp. 228–236, 2009.

[29] P. Gadue, T. L. Huber, P. J. Paddison, and G. M. Keller, "Wnt and tgf-β signaling are required for the induction of an in vitro model of primitive streak formation using embryonic stem cells," *Proceedings of the National Academy of Sciences*, vol. 103, no. 45, pp. 16 806–16 811, 2006.

[30] A. H. Hart, L. Hartley, K. Sourris, E. S. Stadler, R. Li, E. G. Stanley, P. P. Tam, A. G. Elefanty, and L. Robb, "Mixl1 is required for axial mesendoderm morphogenesis and patterning in the murine embryo," *Development*, vol. 129, no. 15, pp. 3597–3608, 2002.

[31] L. Li, L. Song, C. Liu, J. Chen, G. Peng, R. Wang, P. Liu, K. Tang, J. Rossant, and N. Jing, "Ectodermal progenitors derived from epiblast stem cells by inhibition of nodal signaling," *Journal of molecular cell biology*, vol. 7, no. 5, pp. 455–465, 2015.

[32] R. C. Lindsley, J. G. Gill, M. Kyba, T. L. Murphy, and K. M. Murphy, "Canon-

ical wnt signaling is required for development of embryonic stem cell-derived mesoderm," *Development*, vol. 133, no. 19, pp. 3787–3796, 2006.

[33] S. Tada, T. Era, C. Furusawa, H. Sakurai, S. Nishikawa, M. Kinoshita, K. Nakao, T. Chiba, and S.-I. Nishikawa, "Characterization of mesendoderm: a diverging point of the definitive endoderm and mesoderm in embryonic stem cell differentiation culture," *Development*, vol. 132, no. 19, pp. 4363–4374, 2005.

[34] D. Sinner, S. Rankin, M. Lee, and A. M. Zorn, "Sox17 and β-catenin cooperate to regulate the transcription of endodermal genes," *Development*, vol. 131, no. 13, pp. 3069–3080, 2004.

[35] S. Agnihotri, A. Wolf, D. Picard, C. Hawkins, and A. Guha, "Gata4 is a regulator of astrocyte cell proliferation and apoptosis in the human and murine central nervous system," *Oncogene*, vol. 28, no. 34, pp. 3033–3046, 2009.

[36] K. Ryan, N. Garrett, A. Mitchell, and J. Gurdon, "Eomesodermin, a key early gene in xenopus mesoderm differentiation," *Cell*, vol. 87, no. 6, pp. 989–1000, 1996.

Vita

Ding Ding was born on Oct 21th, 1989 in Yinchuan, China. She attended Shanghai Jiao Tong University where she received her Bachelors degree in Information Engineering in 2010 and her Doctor of Medicine degree in 2014.

Ding started research her first year in the Department of Biostatistics with Professor Hongkai Ji. Her project involved genomic data, mainly RNA-Seq data analyses.

While in graduate school, she served as the teaching assistant for course Biostatistics for Lab Scientist.

Ding DING

615 N. Wolfe Street, Room E3038, Baltimore, MD 21205. USA Email: dding6@jhu.edu

Cell: (667) 225-2368

PROFILE

Researcher of statistical methodology with 2 years of experience in analysis of Clinical and Experiment Data, Genome Data; comfortable with working on novel and large datasets. Experienced consultant for statistical analysis of clinical research. Skilled R and SAS programmer and quick learner.

EDUCATION

2015 –2017 (expected)	**Sc. M. in Biostatistics**, Johns Hopkins Bloomberg School of Public Health (JHSPH), Baltimore, MD 21205
2010 – 2014	**M. D. in Clinical Medicine,** Medical School of Shanghai Jiao Tong University, Shanghai, P. R. China
2006 – 2010	**B. S. in Information Engineering**, Shanghai Jiao Tong University, Shanghai, P. R. China

PROFESSIONAL EXPERIENCE

2015 – present **Research Assistant (Sc. M Student),** Dept. of Biostatistics, JHSPH, Baltimore, MD

- **Statistical Consulting Practice**
  - Collaborated with biologists and physicians to help them solve real-world problems via statistical analysis.
  - Observed at student walk-in clinic at Consulting Center of Department of Biostatistics, JHSPH.
  - Carried out cases from initial consulting projects, cleared data, performed statistical analysis, presented the final report.
- **Spanning Tree Representative in High Dimensional Data Project**
  - Processed raw single cell RNA-Seq data from several development process datasets.
  - Built statistical method to construct trees with adjustable nodes and branches to represent the data with machine learning algorithms.
  - Established a marginal probability score to select the best tree describing the biological process with gene expression data.
- **ARQiv High-Through Output Drug Screening Project**
  - Worked with biologists to develop a new random sampling method for pre-experiment estimation.
  - Transformed the whole experiment computation process to be more efficient and accessible through R package.
  - Developed a user friendly graphic interface of the method to help scientists avoid the computation coding part .
- **HEAP app Project**

- Worked with researchers to develop a web-based evaluation system for Dementia Patients' living condition.
- The app helped researchers to collect data of Dementia Patients interactively and efficiently.
- The app could save the data collected to certain online database and generate local report instantly.

- **Highway Fatal Accident Data Analysis Project**
- Processed Highway Fatal Accident Data from 2003 to 2015 and built a cleaned data set for risk factors.
- Built statistical models to investigate what factors contributed to death in injured people.

2014- 2015	**Residency(Intern) of Internal Medicine,**	Renji Hospital, Shanghai, P. R. China
2013 – 2014	**Internship(Medical Student**),	Thoracic Oncology Center of Shanghai Chest Hospital, Shanghai, P. R. China

PUBLICATIONS

- Norrie JL, Li Q, Co S, Huang BL, **Ding D**, Uy JC**,** Ji ZC, Mackem S, Bedford MT, Galli A, Ji HK**,** Vokes SA (2016) PRMT5 is essential for the maintenance of chondrogenic progenitor cells in the limb bud. ***Development***. To appear
- White DT, Eroglu AU, Wang G, Zhang L, Sengupta S, **Ding D**, Rajpurohit SK, Walker SL, Ji HK, Qian J, Mumm JS (2016) ARQiv-HTS, a versatile whole-organism screening platform enabling in vivo drug discovery at high-throughput rates. ***Nature Protocols*** 11:2432–2453
- Sun, S.Q., Zhang, T., **Ding, D.,** Zhang, W.F., Wang, X.L., Sun, Z., Hu, L.H., Qin, S.Y., Shen, L.H. and He, B.,(2016). Circulating MicroRNA- 188,- 30a, and- 30e as Early Biomarkers for Contrast- Induced Acute Kidney Injury***. Journal of the American Heart Association, 5(8)***, p.e004138.
- Zhang, W.F., Zhang, T**., Ding, D**., Sun, S.Q., Wang, X.L., Chu, S.C., Shen, L.H. and He, B.,(2017). Use of Both Serum Cystatin C and Creatinine as Diagnostic Criteria for Contrast- Induced Acute Kidney Injury and Its Clinical Implications. ***Journal of the American Heart Association, 6(1***), p.e004747.
- **Ding, D**., Yu, Y., Li, Z., Niu, X. and Lu, S., (2014). The predictive role of pretreatment epidermal growth factor receptor T790M mutation on the progression-free survival of tyrosine-kinase inhibitor-treated non-small cell lung cancer patients: a meta-analysis***. Onco Targets Ther***, 7, pp.387-93.

COMPUTING SKILL

Programming Language:

Proficient in R, SAS, Matlab and C;

Functional in STATA and C++

SAS Certified Base Programmer for SAS 9